计算机与信息工程学院实验报告

姓名： 学号：专业： 计算机科学与技术 年级： 2020

课程： 移动应用开发技术 主讲教师： 陈小潘 辅导教师： 陈小潘

实验时间： 2022 年 3 月 30 日 上 午 10 时至 12 时，实验地点 606实验室

**实验题目**：实验 2 Android 用户界面

**实验目的**：1.掌握各种界面控件的使用方法 。2. 掌握各种界面布局的特点和使用方法 3. 掌握各种对话框的特点和使用方法

**实验环境**（硬件和软件） Windows 7/Android Studio

**实验内容：**

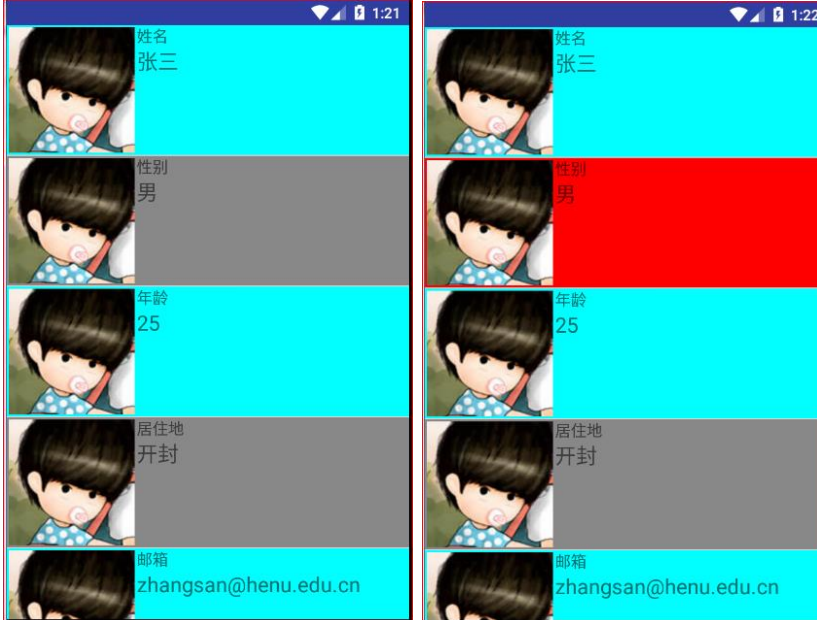
1. 使用 ListView 实现如下程序界面和功能，如图 4 所示。



当选择某一 ListView 子项，TextView（即 MenuDemo 位置）中会显示所选 子项内容，如图 5 所示



2、采用自定义布局 BaseAdapter 修改列表颜色，如图 6 所示。

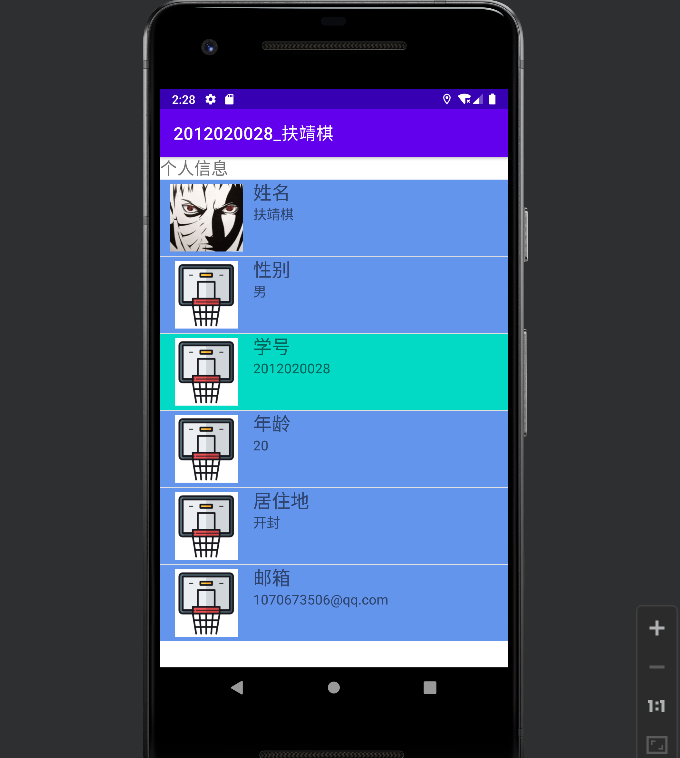


**实验步骤：**

1. **编写继承与Activity的入口界面类-MainActivity**
2. **编写自己的Adapter，继承与BaseAdapter的MyAdapter**
3. **编写布局文件ListView主文件activity\_main.xml**
4. **编写列表项布局文件list\_view.xml**

**实验数据记录：**

1. **实验截图**



核心代码：（监听器实现基本交互功能）

1. 实现列表各个表项，同时设置监听与点击事件相关联。

list = new ArrayList<Map<String,Object>>();

        listView = (ListView) findViewById(R.id.listView);

        Map<String,Object> map1 = new HashMap<String,Object>();

        Map<String,Object> map2 = new HashMap<String,Object>();

        Map<String,Object> map3 = new HashMap<String,Object>();

        Map<String,Object> map4 = new HashMap<String,Object>();

        Map<String,Object> map5 = new HashMap<String,Object>();

        Map<String,Object> map6 = new HashMap<String,Object>();

        map1.put("title","姓名");

        map1.put("info","扶靖棋");

        map1.put("picture",R.drawable.img\_1);

        list.add(map1);

        map2.put("title","性别");

        map2.put("info","男");

        map2.put("picture",R.drawable.img);

        list.add(map2);

        map3.put("title","学号");

        map3.put("info","2012020028");

        map3.put("picture",R.drawable.img);

        list.add(map3);

        map4.put("title","年龄");

        map4.put("info","20");

        map4.put("picture",R.drawable.img);

        list.add(map4);

        map5.put("title","居住地");

        map5.put("info","开封");

        map5.put("picture",R.drawable.img);

        list.add(map5);

        map6.put("title","邮箱");

        map6.put("info","1070673506@qq.com");

        map6.put("picture",R.drawable.img);

        list.add(map6);

        MyAdapter adapter = new MyAdapter(this,list,R.layout.user);

        listView.setAdapter(adapter);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override

            public *void* onItemClick(AdapterView<?> adapterView, View view, *int* i, *long* l) {

                adapter.setCurrentItem(i);

                adapter.setClick(true);

                adapter.notifyDataSetChanged();

            }

        });

1. 给xml布局文件创建java对象，获取数据显示在各组件上

public View getView(*int* i, View view, ViewGroup viewGroup) {

            // 声明内部类

            Util util = null;

            final *int* flag = i;

            if(view == null){

                util = new Util();

                LayoutInflater inflater = LayoutInflater.from(context);

                view = inflater.inflate(resource, null);

                // 指向布局文件内部组件

                util.picture = (ImageView) view.findViewById(R.id.picture);

                util.info = (TextView) view.findViewById(R.id.info);

                util.title = (TextView) view.findViewById(R.id.title);

                // 增加额外变量

                view.setTag(util);

            }

            else {

                util = (Util) view.getTag();

            }

            Map<String, Object> map = list.get(i);

            util.info.setText((String) map.get("info"));

            util.title.setText((String) map.get("title"));

            util.picture.setImageResource((Integer) map.get("picture"));

            LinearLayout parent= (LinearLayout) view.findViewById(R.id.parent1);

            if (mCurrentItem==i&&isClick){

                parent.setBackgroundColor(Color.parseColor("#FF03DAC5"));

            }else{

                parent.setBackgroundColor(Color.parseColor("#6495ED"));

            }

            return view;

        }

**问题讨论（此部分为重点书写内容）：**

**问题一：** 选择Adapter实现任务功能，复杂的布局与点击变换颜色功能无法实现。

**解决方法：**

BaseAdapter（自定义的时候很多，尤其是一些复杂的布局，不自定义实现不了啊）：实现复杂的列表布局，由于BaseAdapter是一个抽象类，使用该类需要自己写一个适配器继承该类，正是由于继承了该类，需要我们重写一些方法，让我们可以在代码里控制列表的样式，更加灵活。即自己写一个适配器。

代码思路如下：

**重写getCount(),getElem()，getView()等函数，如果你的getCount()返回值是0的话，列表将不显示同样return 1，就只显示一行。**  
}

        @Override

        public *int* getCount() {

            return list.size();

        }

        @Override

        public Object getItem(*int* i) {

            return list.get(i);

        }

        @Override

        public *long* getItemId(*int* i) {

            return i;

        }

问题解决。

**问题二：** 列表显示布局时，无法正常显示各项数据，程序闪退。

**解决方法：**

显示列表时，首先实例化一个适配器（这里将实例化自定义的适配器）。当手动完成适配时，必须手动映射数据，这需要重写getView（）方法。getView()有三个参数，position表示将显示的是第几行，covertView是从布局文件中inflate来的布局。

@Override

        public View getView(*int* i, View view, ViewGroup viewGroup) {

            // 声明内部类

            Util util = null;

            final *int* flag = i;

            if(view == null){

                util = new Util();

                // 给xml布局文件创建java对象

                LayoutInflater inflater = LayoutInflater.from(context);

                view = inflater.inflate(resource, null);

                // 指向布局文件内部组件

                util.picture = (ImageView) view.findViewById(R.id.picture);

                util.info = (TextView) view.findViewById(R.id.info);

                util.title = (TextView) view.findViewById(R.id.title);

                // 增加额外变量

                view.setTag(util);

            }

            else {

                util = (Util) view.getTag();

            }

            // 获取数据显示在各组件

            Map<String, Object> map = list.get(i);

            util.info.setText((String) map.get("info"));

            util.title.setText((String) map.get("title"));

            util.picture.setImageResource((Integer) map.get("picture"));

            LinearLayout parent= (LinearLayout) view.findViewById(R.id.parent1);

            if (mCurrentItem==i&&isClick){

                parent.setBackgroundColor(Color.parseColor("#FF03DAC5"));

            }else{

                parent.setBackgroundColor(Color.parseColor("#6495ED"));

            }

            return view;

        }

**问题三：** 如何在drawable目录下插入图片

**解决方法：**

一种方法：直接将已知图片粘到该目录下，同时为了将其引用，通过代码

map1.put("picture",R.drawable.*img\_1*);

寻找图片，并插入到各项数据中。

另一种方法：

String imageName = "picture"

*int* resID = getResources().getIdentifier(imageName, "drawable", "package.name");

ImageView image;

image.setImageResource(resID );

能够引用图像资源并将ImageView设置为它.